BERKELEY LAB

Net-Preflight Check: Using File transfer to Measure Network Performance before Large Data Transfers

Bashir Mohammed [1], Mariam Kiran[2], Bjoern Enders[3]
Lawrence Berkeley National Laboratory, Berkeley, California, USA

[1] Scientific Data Management, (SDM), [2] Energy Sciences Network (ESnet), [3]National Energy Research Scientific Computing Center (NERSC)

Contact Email: Bmohammed@lbl.gov

The Sixteenth International Conference on Systems and Networks Communications
ICSNC 2021
October 03, 2021 to October 07, 2021

IARIA

Presenter's Bio

Bashir Mohammed received his master's degree in Control Systems from the University of Sheffield UK, and his PhD from the University of Bradford UK. He is currently a Postdoctoral research fellow at Lawrence Berkeley National Lab in Berkeley California, where is working on the Deep Learning and Artificial Intelligence High-Performance Network (DAPHNE) project .

His research interest lies at the intersection between Network Systems, Control Systems, and Machine Learning.

Talk Plan

- Motivation from Real World
- Current State of the art, and Objectives
- Methodology
- Use Cases
- Experimental Design and Results
- Summary and Conclusion

Motivation from Real World

- Challenge for Scientists before doing a transfer (Eg. DOE facilities):
  - Sometimes network performance is degraded and we don't know why
  - Where is the problem? the responsible agent (“point the finger”) in a multi-agent problem **confusing** the scientist even more.
  - opening a ticket to troubleshoot **incurs an intolerable delay**

- There is demand for a tool:
  - That can be launched before or during an experiment
  - That delivers **metrics relevant to bulk data** transfers in a pleasing form and compare performance before and now

- NetPreflight can check **network reliability and help find bad configurations**

Current State of the art

- **To get accurate throughput measurement between two network points, you need to install and run a network monitoring tool in a client-to-server mode.**
  - You need to make an installation in both source and destination point/server.

- **Lack of memory to say if config has changed or how performance has been in the past**
  - Memory file which stores previous configurations and network performances with different file transfers and time stamps.

- **Lack of available throughput measurement during bulk data transfer for exascale scientific applications.**
  - Network weather monitoring tool for very large concurrent data transfer not very common.

Objective

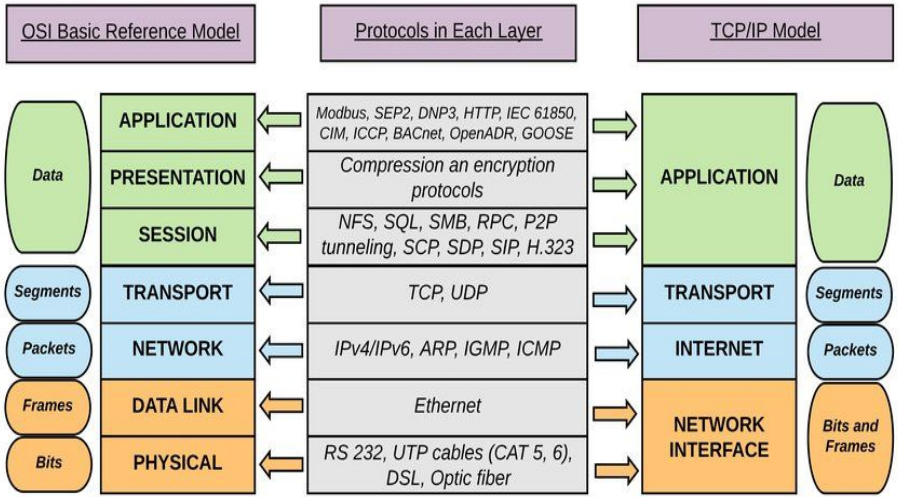
- To develop a custom, simple end-to-end, lightweight monitoring tool for measuring available throughput on a well provisioned network without a client-to-server mode setup.

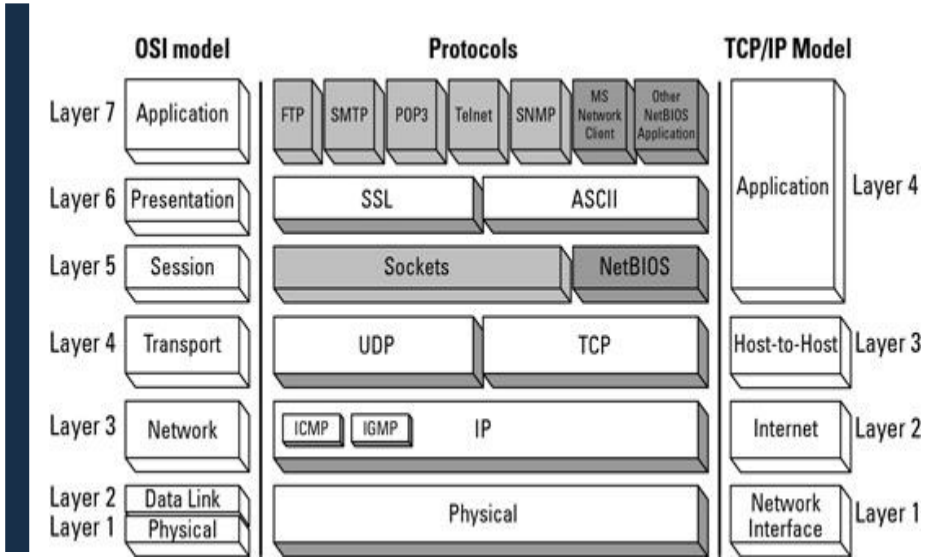
- We compare with existing tools focusing on throughput accuracy, security, implementation and flexibility.

- To test the tool's performance in different network setups like isolated network and public Internet network.

Methodology

- We utilized Sockets and the actual file size to measure the throughput between two points .
- They are programming schema in which sockets are used and manipulated to create a connection between softwares. Located in between L3 and L4 TCP/IP conceptual layers.





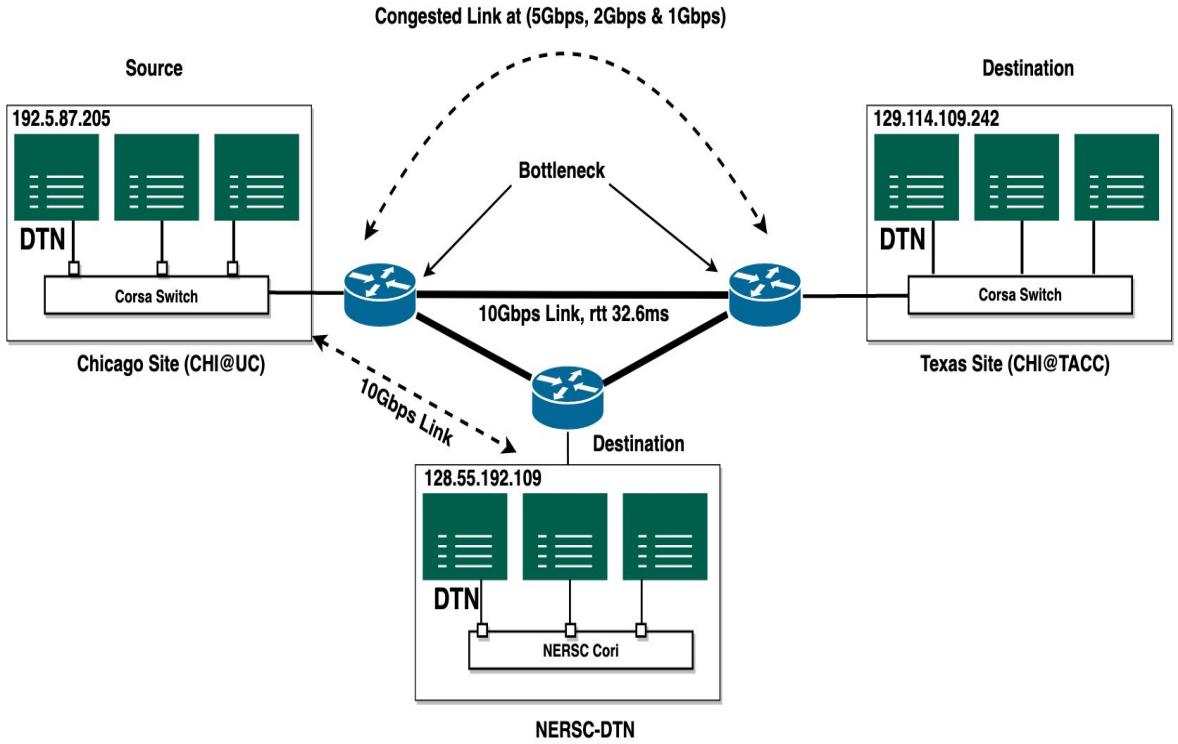
$$T_p = \frac{B}{\Delta T} \quad , \quad \Delta T = t_n - t_s$$

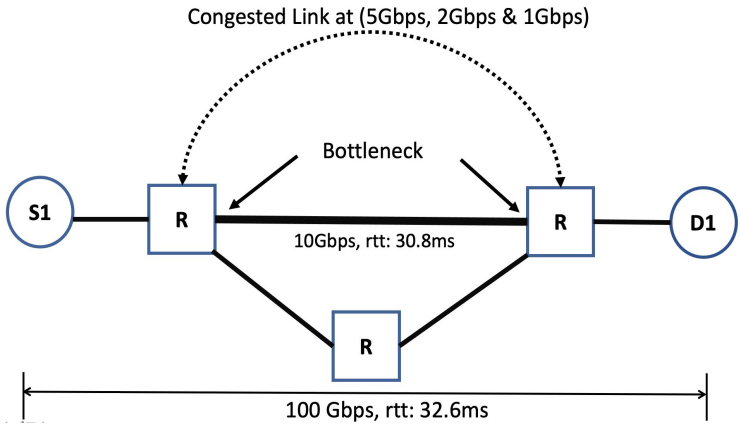
$$\frac{\text{TCP-Window-Size-in-bits}}{\text{Latency(sec)}} = \text{TP(Bits/sec)}$$

https://www.dummies.com/programming/networking/cisco/network-basics-tcpip-and-osi-network-model-comparisons/

Use Cases

- Isolated Network (Network Setup between (CHI@UC toCHI@TACC)
- Public Internet (Network Setup between (NERSC DTN - CHI@UC)





Experiments and Results

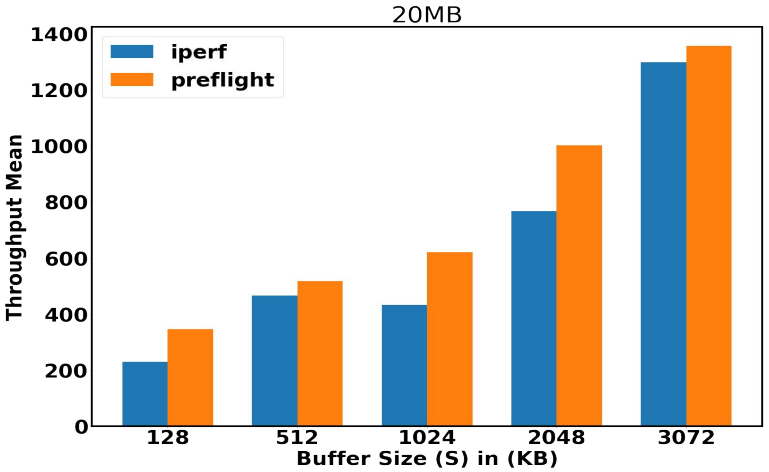
- **Experiment 1- Measurement w.r.t Buffer Size and File Size at 10 Gbps link Capacity**
- **Experiment 2 - Measurement w.r.t different TCP congestion algorithm with 10Gbps link Capacity**
- **Experiment 3- Aggregated Throughput Measurement limiting bandwidth w.r.t Congested Vs Non-congested links**
- **Experiment 4 - Tool Comparison w.r.t Window size and data transfer, iperf Vs preflight**

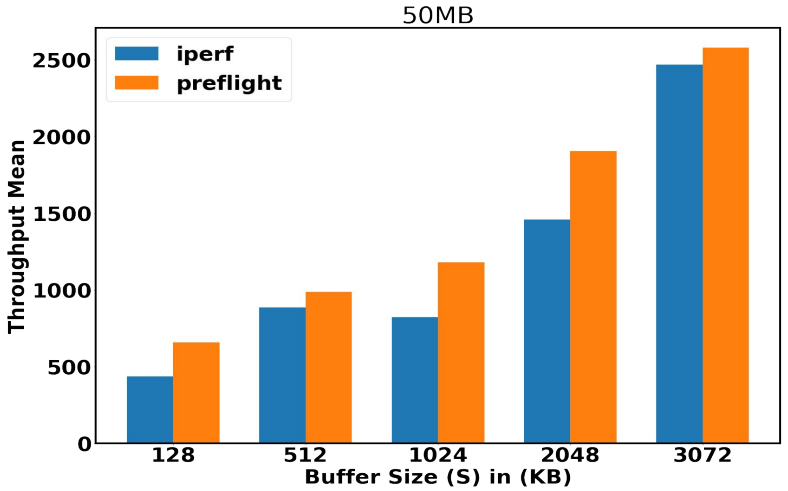
Test Procedures and Results in memory

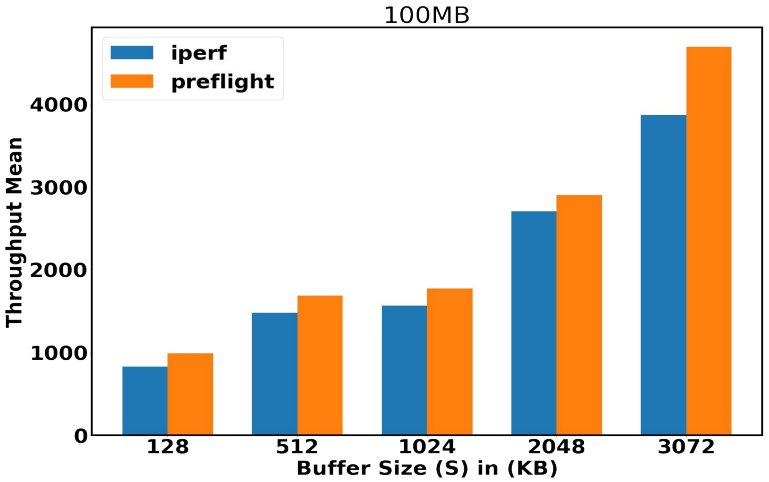
- **E.g** `python <scriptname> -H <TargetHostIPaddress> -K <KeyFilepath> -F <targetFilepath> -I <no. of iterations> -S <tagetFilesize>`

```
[mohammed@dtn01 ~/experiments]$ cat memory.json
{
    "192.5.86.216": {
        "iteration": 2,
        "Throughput": 479.06153,
        "lapse": 0.171,
        "BufferSize": 1024,
        "Timestamp": "2021-03-09 04:04:12.851378",
        "trace_result": [
            "traceroute to 192.5.86.216 (192.5.86.216), 30 hops max, 60 byte packets\n",
            " 1  vip-205.nersc.gov (128.55.205.1)  0.193 ms vr205-cr2.nersc.gov (128.55.205.254)  0.197 ms  0.219 ms\n",
            " 2  cr2-br1.nersc.gov (128.55.192.113)  0.612 ms cr1-br1.nersc.gov (128.55.192.109)  0.589 ms  0.776 ms\n",
            " 3  sunncr5-nersc.es.net (198.129.78.33)  1.596 ms  1.736 ms  1.616 ms\n",
            " 4  sacrcr55-ip-b-sunncr55.es.net (134.55.40.145)  5.450 ms  5.339 ms  5.348 ms\n",
            " 5  chiccr55-ip-a-sacrcr55.es.net (134.55.40.150)  46.871 ms  46.879 ms  46.856 ms\n",
            " 6  anl-ip-g-chiccr5.es.net (198.124.216.234)  48.951 ms  48.962 ms  48.942 ms\n",
            " 7  192.5.86.216 (192.5.86.216)  49.209 ms  49.262 ms  49.246 ms\n"
        ]
    },
    "129.114.109.229": {
        "iteration": 1,
        "Throughput": 435.74235999999996,
        "lapse": 0.188,
        "BufferSize": 1024,
        "Timestamp": "2021-03-09 03:50:44.426669",
        "trace_result": [
            "traceroute to 129.114.109.229 (129.114.109.229), 30 hops max, 60 byte packets\n",
            " 1  vr205-cr2.nersc.gov (128.55.205.254)  0.193 ms  0.228 ms  0.228 ms\n",
            " 2  cr2-br1.nersc.gov (128.55.192.113)  0.457 ms  0.727 ms  0.974 ms\n",
            " 3  sunncr5-nersc.es.net (198.129.78.33)  1.668 ms  1.653 ms  1.642 ms\n",
            " 4  elpacr5-ip-a-sunncr5.es.net (134.55.37.42)  25.426 ms  25.531 ms  25.459 ms\n",
            " 5  houscr5-ip-a-elpacr5.es.net (134.55.40.198)  39.757 ms  39.800 ms  39.830 ms\n",
            " 6  learn-houscr5.es.net (198.129.33.162)  39.979 ms  39.998 ms  39.979 ms\n",
            " 7  tx-bb-i2-hstn.tx-learn.net (74.200.187.26)  41.179 ms  41.083 ms  41.065 ms\n",
            " 8  192.88.12.234 (192.88.12.234)  49.453 ms  48.850 ms  48.833 ms\n",
            " 9  qfx10008-vl661-ptp.net.tacc.utexas.edu (129.114.0.141)  45.315 ms  45.252 ms  45.688 ms\n",
            "10  chi-dyn-129-114-109-229.tacc.chameleoncloud.org (129.114.109.229)  44.553 ms  44.732 ms  44.679 ms\n"
        ]
    }
```

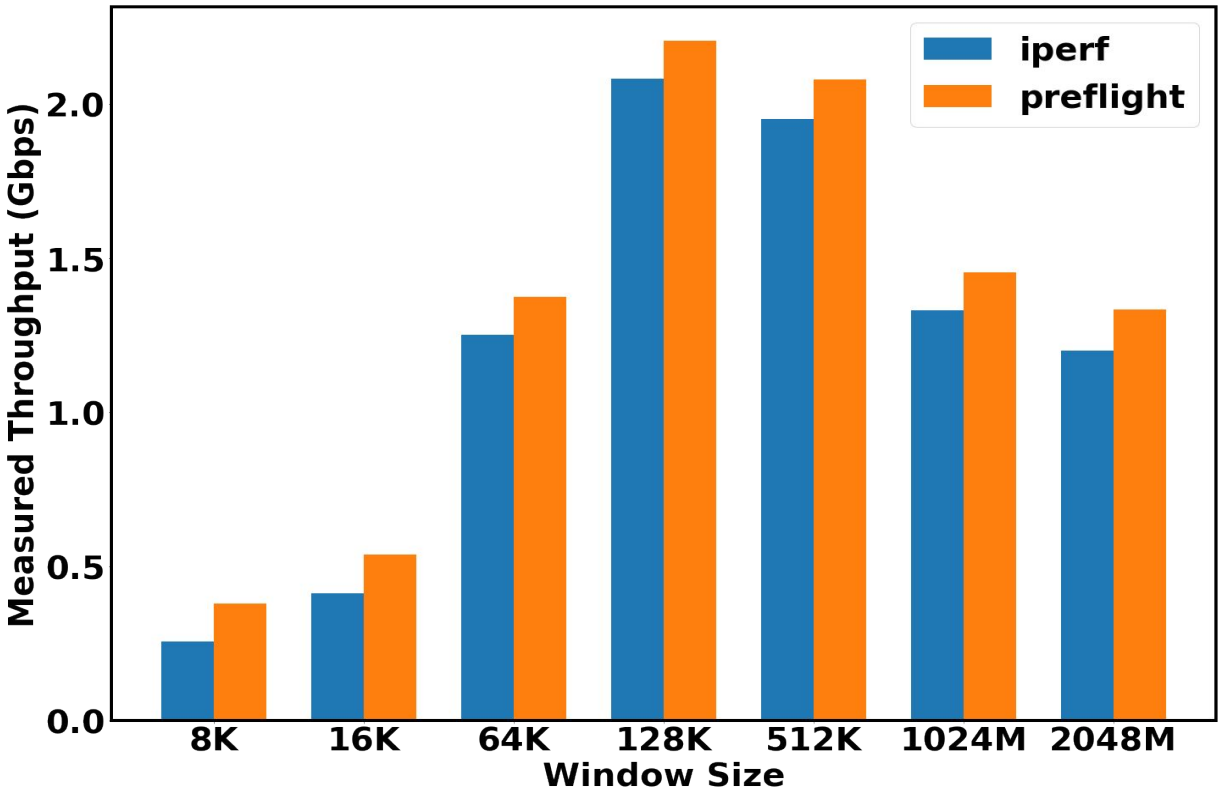
Preliminary Results - Chicago to Texas Chameleon Testbed



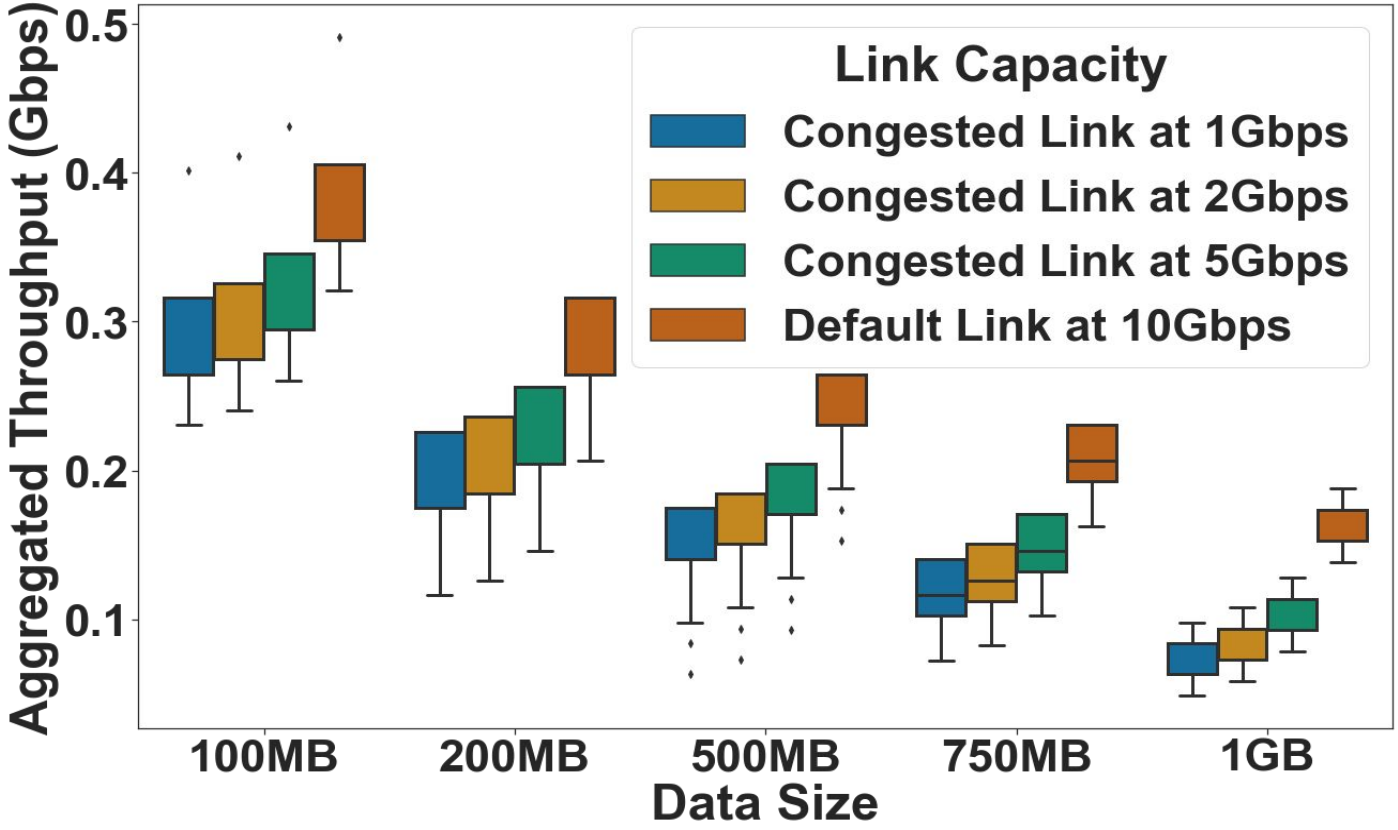




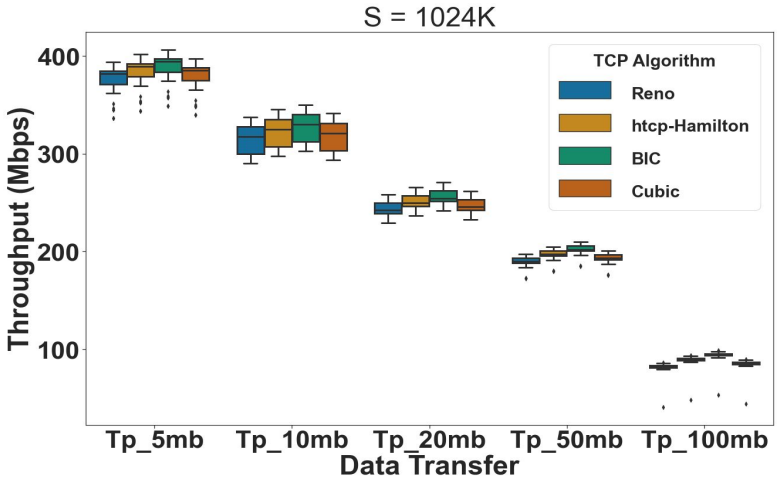
Results - Effect of window size variation on performance.

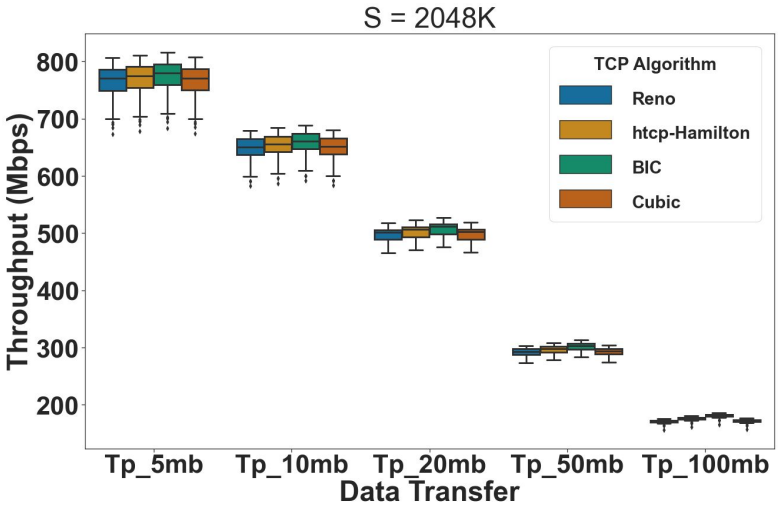


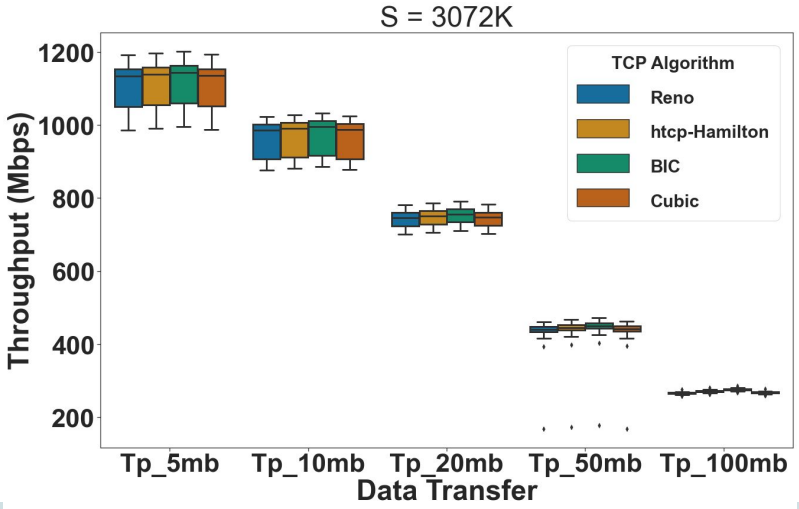
Results - Bottleneck link capacity



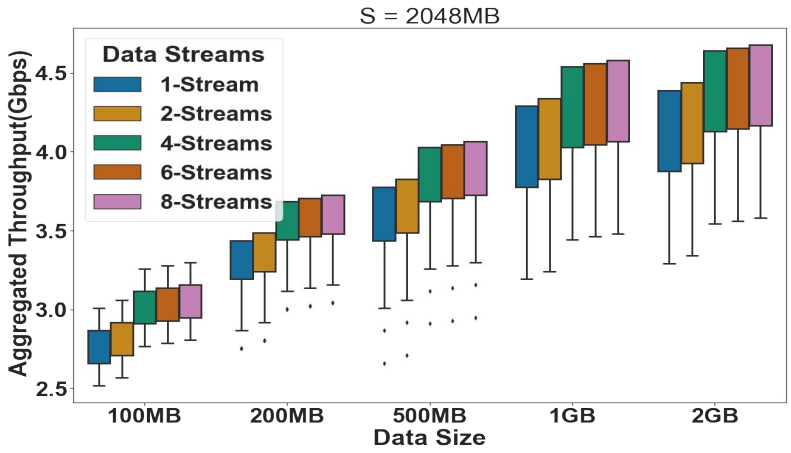
Results - Using Different TCP CC algo. from CHI-TACC

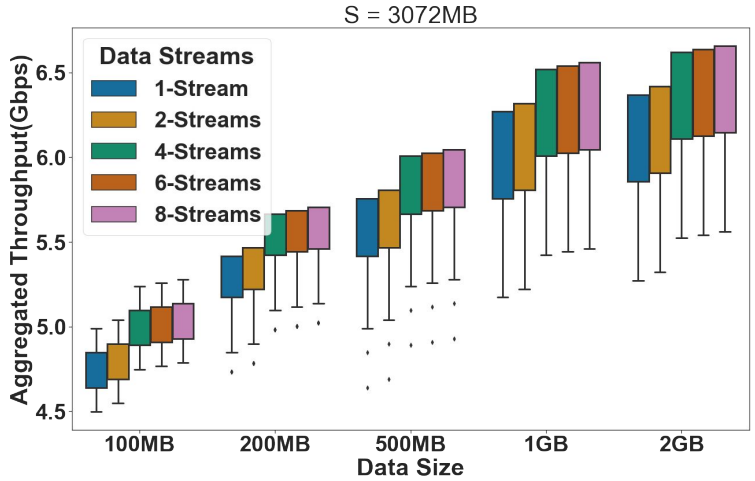


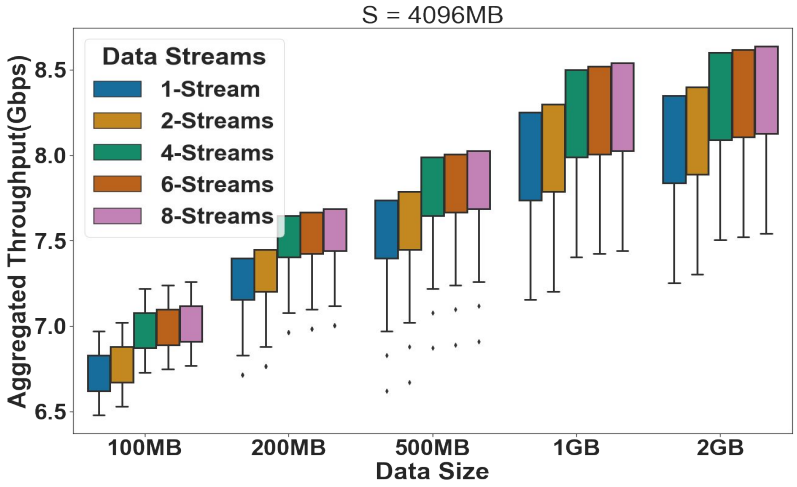




Results:Concurrent Multiple transfers NERSC-CHICAGO







Summary and Conclusion

- We introduced a simple end-to-end, light-weight tool for measuring available throughput. It comes with a memory which stores previous configurations and network performances
- It addresses the question of how concurrent stream connections can improve aggregate TCP throughput measurement .
- It also addresses the question of how to select the maximum number of sockets necessary to maximize TCP throughput while simultaneously avoiding congestion
- A theoretical model was developed to analyze the questions. It was validated by a series of experiments.
- Our findings indicate that in the absence of congestion, the use of parallel TCP connections is equivalent to using a large. In the future we will continue to improve the performance and integrate it popular Grid data transfer applications.

Thanks for listening!
Questions???

Email: bmohammed@lbl.gov